X-711-67-459

NASA TM X- 55940

AN LSI VARIABLE FUNCTION REGISTER

SEPTEMBER 1967

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

X-711-67-459

AN LSI VARIABLE FUNCTION REGISTER

Robert J. Lesniewski
Flight Data Systems Branch
Spacecraft Technology Division

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

ABSTRACT

A variable function register capable of ten different operations is presently being developed through MOS technology. The register, which uses 500 active devices, can be cascaded to form a large composite register which is capable of functioning as one large register or as individual registers. This is made possible by electrically altering the characteristics of the register. This allows the variable function register to meet various needs.

PRECEDING PAGE BLANK NOT FILMED.

CONTENTS

ILLUSTRATIONS

TABLES

AN LSI VARIABLE FUNCTION REGISTER

INTRODUCTION

Since spacecraft experiments are continually increasing in sophistication, the need to do on-board data processing will soon reach a point of absolute necessity. Large Scale Integration (LSI) will play a major role in providing the required circuitry to do this on-board processing.

One present drawback to LSI systems or subsystems is their tendency to be specialized. A given LSI shift register, for example, can shift only in one direction even though it contains hundreds of active elements. This limitation and specialization of complex circuitry seems inefficient. Hundreds of active elements configured to do just one job defeat the true value of LSI.

This paper will show how LSI can be developed to meet many anticipated applications instead of specialized one user, one application type of arrays. An example is the Variable Function Register. The design of this register includes an electrical reconfiguration capability and a repertoire of ten operations geared to provide the user with a flexible device. A second generation register with three times the capability will be briefly mentioned as a further example of how LSI can provide the user with a multi-purpose multi-function device.

VARIABLE FUNCTION REGISTER

The variable function register is basically a parallel-serial shift register with a fixed length. The distinction between this register and others is not only its instruction set but also its ability to modify the manner of the instruction execution.

The register is capable of executing the following operations:

1. Parallel loading of data into the register.
2. Parallel outputting of data from the register.
3. Shifting of data to the right.
4. Shifting of data to the left.
5. Rotating data to the right.
6. One's complementing the contents of the register.
7. Clearing simultaneously the register to zero.

In addition to this basic repertoire of seven distinct operations, the variable function register has an electrically alterable capability, whereby three of the basic seven operations can be modified to meet similar but different requirements of the register. The three operations in question involve the shift capability. By altering the shift operations, identical registers can be cascaded to form extended precision registers of any length required. (The only limitation is the restriction of the total bit length to multiples of the basic register.)

Were it not for complementary Metal Oxide Silicon (MOS) technology such a device would not be practical from a power dissipation and an element count point of view. MOS technology offered the most feasible as well as practical method of obtaining high device density compared to current bipolar technology. However, while MOS technology in general offered low power dissipation, it was felt that the lowest power dissipation could be best had by using complementary MOS technology.

Why Variable Function

The reasons for designing a variable function register came as an evolution of thought. It was noticed that two registers of equal length but slightly different operating modes were required in a system design. Each register with its control was to be integrated on a chip. Looking at the similarity of the two registers made it clear that it would be cheaper to develop one register with the ability to do both jobs than it was to develop two slightly different registers. Further evolution of these ideas led to the decision to build in sufficient controls to alter the basic register to meet any one of the four different requirements that existed. Because of the register's versatility, the entire system was more flexible since it could be expanded without changing device designs.

Not only was there a savings in cost but the number of different types of register chips was reduced four to one. From this, the concept of electrical reconfiguration (modes) was developed.

Figure 1 shows the basic circuitry of the shift register stage excluding its interconnection circuitry with other stages. It is the interconnection design which makes it possible to convert a number of register stages into a variable function register. Before going into any details, let us first define what functions a shift register should have.

PROPERTIES OF A GENERAL REGISTER

A shift register should be capable of shifting either to the right or left, or shifting to the right in a continuous loop. If a register shifts to the right, the following can occur at both ends of the register:

BASIC REGISTER STAGE

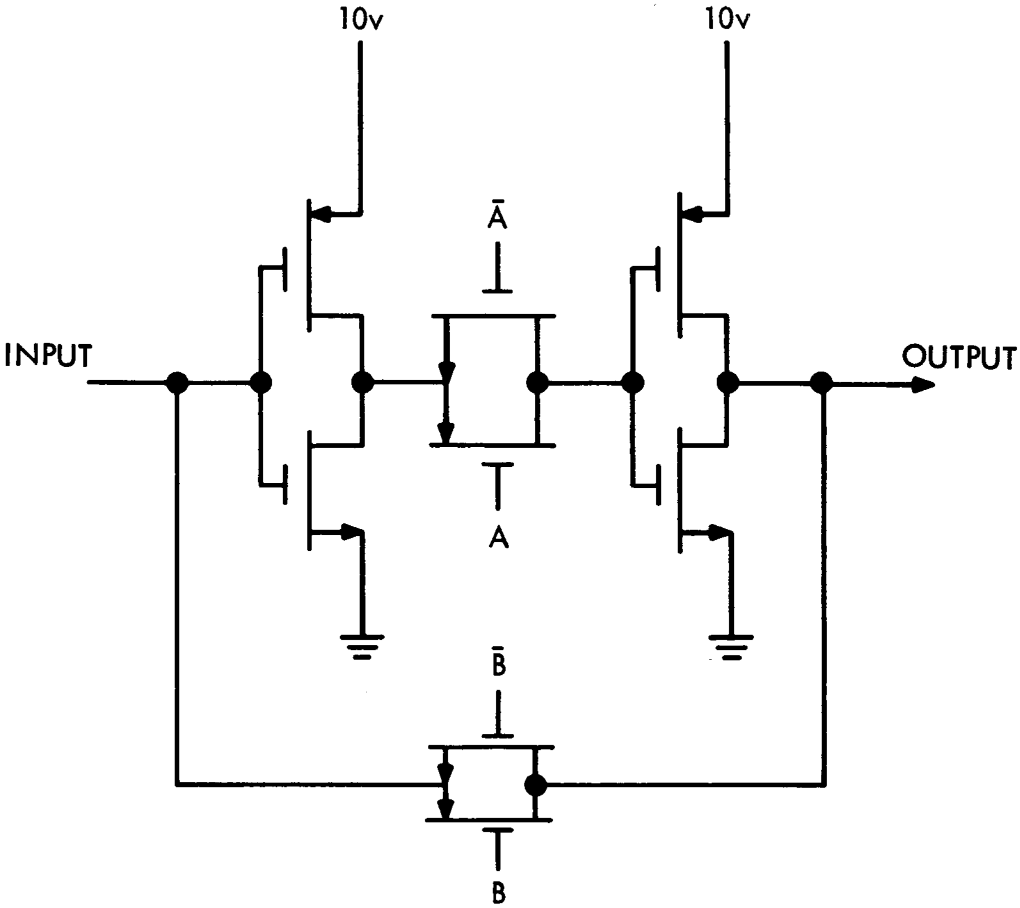


Figure 1

1. The right end will not output serial data

   or

2. The right end will output serial data
3. The left end will accept serial data

   or

4. The left end will not accept serial data and,
   a. fill zeroes in from the left to the right
   b. fill the contents of the most significant stage from left to right.

If a register shifts to the left, the following can occur at both ends of the register:

1. The right end will accept serial data

   or

2. The right end will not accept serial data and feed zeroes from right to left.
3. The left end will output serial data

or

4. The left end will not output serial data.

If the register serially rotates to the right, the least significant register stage, which is on the right, shifts data to the most significant register stage, which is on the left. And, if two or more registers are rotating to the right as one large register, each register should accept serial data at the left end and output serial data at the right end.

A register could also have a bypass line where serial data could shunt around a register without disturbing its contents.

PROPERTIES OF VARIABLE FUNCTION REGISTER

The variable function register is capable of non-total shifting to the right or left, and rotating to the right. The most significant bit (MSB) in a right shift is forced to either zero or it may not change state. Thus, a non-total right shift either propagates to the right zeroes or only the value of the most significant bit. The choice of either propagation is left up to the designer or programmer. Propagating the most significant bit to the right is an operation which is compatible to two's complement arithmetic schemes, where the sign is propagated.

The non-total left shift is similar to the non-total right shift in transfer of data except data is shifted left while zeroes are propagated from right to left filling vacated bit positions.

The non-total rotate shift performs a serial transfer of data to the right in the form of a closed loop. The rotate allows each cell to transfer data to its neighbor on the right. The least significant stage transfers data to the most significant stage.

Thus, the register is capable of shifting to the right in two ways: propagating to the right either zeroes or the most significant bit, shifting to the left and rotating to the right. BY DEFINITION, NON-TOTAL SERIAL SHIFTS AFFECT ALL STAGES OF THE REGISTER; HOWEVER, THERE ARE NO SERIAL INPUTS OR OUTPUTS OF THE REGISTER. That is, an observer looking at the data lines of the register will notice that before, during and after a non-total shift lines are effectively open circuited. The three non-total shifts are, therefore, considered internal shifts.

The register can be cascaded to form a larger register by electrically modifying in four ways the shift operations of each register of the composite register. These modifications enable serial data to leave, or to enter, or to leave and enter, or to neither leave nor enter the register. The modifications are called modes, and are essentially the properties of total shifts. Therefore, A TOTAL SHIFT WILL TRANSFER SERIAL DATA IN OR OUT OF THE REGISTER IN A MANNER DICTATED BY THE MODE. A total shift occurs if non-total Right, Left, or Rotate instruction is accompanied by the Total instruction. Thus non-total Right and Total together produce a total shift to the right.

Mode 1

A register in Mode 1 (Figure 2-a) is capable of serially communicating with a register connected to its left data line. Serial data is permitted to either transfer in or out of the register on this line during either a total right shift or total left shift. A total right shift permits the most significant bit to accept external data. A non-total right shift will either propagate zeroes or the value of the most significant bit (MSB), which is the left most bit. The left total shift and non-total left shift will propagate zeroes from the least significant bit (LSB), which is the right most bit. External serial data cannot enter or leave on the right data line during either a total right or left shift. Diagonal markings are used to denote those sides in which serial transfers due to total right or left shifts are blocked.

Mode 2

A register in Mode 2 (Figure 2-b) is capable of serially communicating with a register connected to its right data line. A total right or left shift is permitted, respectively, to transfer serial data out or in of this line. However, serial data is not permitted to enter or leave on the left data line. A total right shift in Mode 2 has the same effect on the left side of the register as a non-total right shift. That is either zeroes or the contents of the most significant bit will be propagated to the right. The non-total left shift in Mode 2 operates as in Mode 1.

Mode 3

A register in Mode 3 (Figure 2-c) is capable of serially communicating with a register connected on either the right or left data line during either a total right or left shift. Thus data may serially shift through the register without being altered or blocked. If three registers were cascaded to form one big register would be in Mode 3. It should again be noted that during any non-total shift there is no external transfer of data on either serial line.

MODES

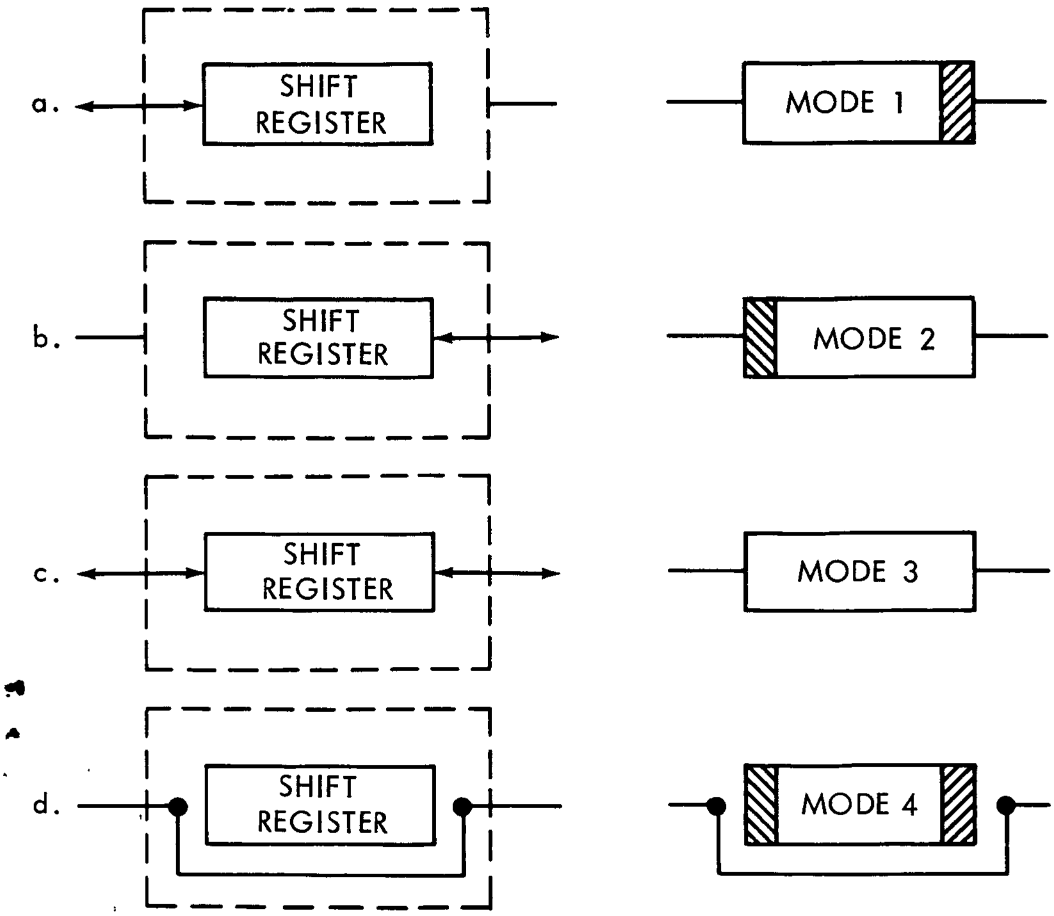


Figure 2

Mode 4

A register in Mode 4 (Figure 2-d) is incapable of communicating with any other register regardless of the shift instructions. A total right or left shift has the same effect as a non-total right or left shift. Therefore, the register is completely isolated electrically from other registers. However, serial data may by-pass the register through a switch which internally connects the left data line with the right data line. Thus a register on either side of a register in Mode 4 can communicate serially through the by-pass switch. While Mode 4 isolates the register, internal shifts may take place within the register, such as a non-total rotate. In fact, regardless of mode the register can internally rotate data from the right end of the register to the left.

The total rotate instruction is similar to the total right shift in Mode 3. The important difference is that a register performing a total rotate in either of Modes 1 or 2 functions like a register performing a total right shift in Mode 3. This means data (Figure 3-b) can enter serially on the left data line of the register, shift through the register and leave on the right data line. However, if the register is in Mode 4 (Figure 3-a), the total rotate instruction itself is inhibited; therefore, no shift occurs. But the by-pass switch is closed permitting registers adjacent to a register in Mode 4 to communicate.

Taking all the shift instructions with the concept of modes yields the same properties associated with a general register with the exception that the variable function register can alter its properties to meet varied requirements.

CONFIGURATIONS

Since the variable function register has four modes of operation along with a choice of propagating to the right the most significant bit, the register can be

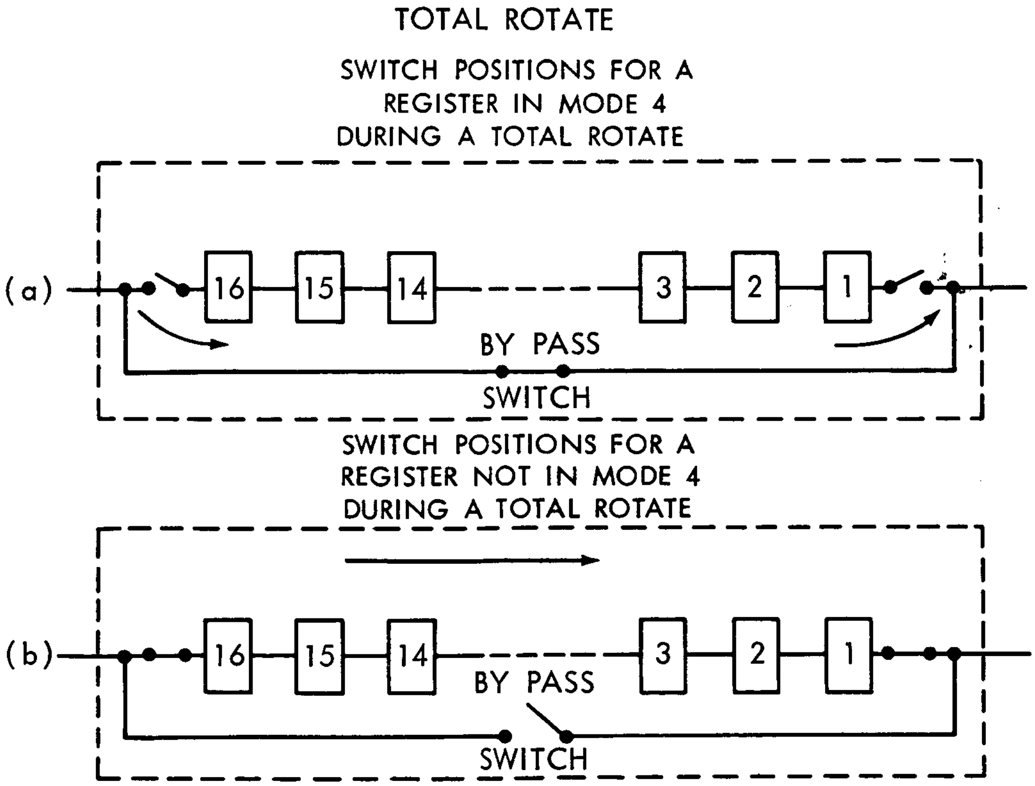

TOTAL ROTATE
SWITCH POSITIONS FOR A REGISTER IN MODE 4 DURING A TOTAL ROTATE
(a)
16
15
14
3
2
1
BY PASS
SWITCH
SWITCH POSITIONS FOR A REGISTER NOT IN MODE 4 DURING A TOTAL ROTATE
(b)
16
15
14
3
2
1
BY PASS
SWITCH

Figure 3

electrically modified in eight different ways. Thus if two registers are cascaded, the number of ways the pair can be modified is $(8)^2$ or 64. In general the combinations increase by $(8)^N$ where N is the number of registers in cascade.

To accomplish the four modes, lines C_1 and C_2 are required as shown in Figure 7. The modes are indicated as follows:

Mode	C_1	C_2
1	0	1
2	1	0
3	1	1
4	0	0

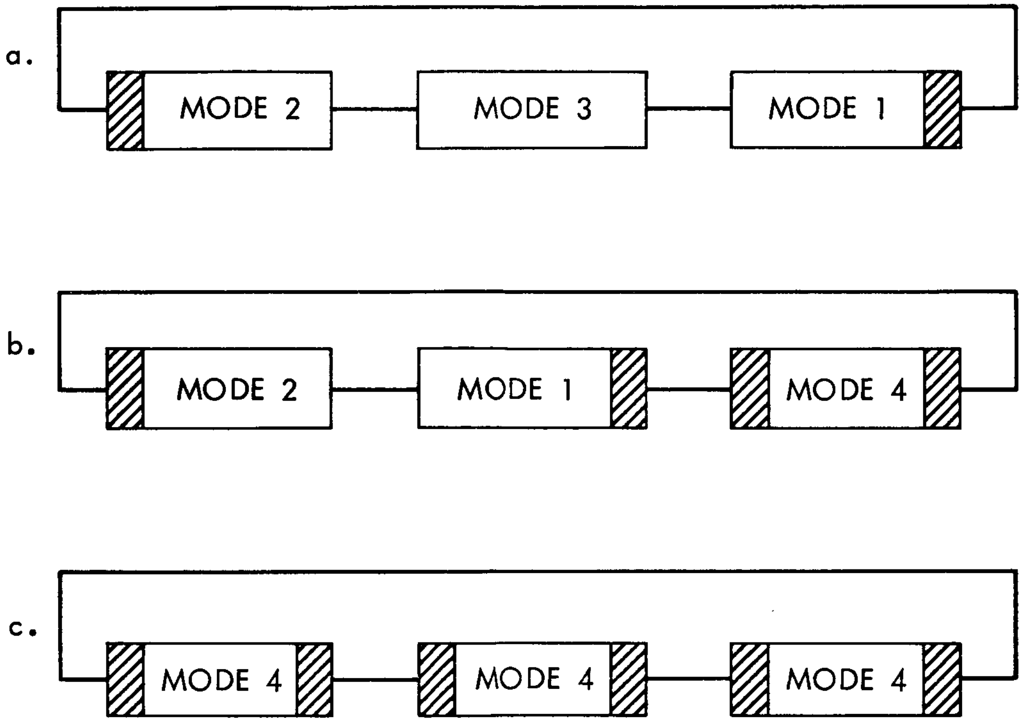
To enable one's or two's complement capability the line C_3 is required. If C_3 is 0 during a right shift the most significant bit will shift its contents from left to right. If C_3 is 1 during a right shift, a zero will be shifted from left to right.

Figure 4 shows just three combinations of a potential $(8)^3$. Figure 4-a is a 48-bit register made up of three 16-bit registers. The register is isolated on either end so that there is no serial transfer of data between the most and least significant bits during a total right or left shift. Since there is a potential feedback loop from the right register to the left register, the entire composite 48-bit register can be rotated to the right during a total rotate. Figure 4-b shows the same three registers, only the right register is in Mode 4 (isolated). The two left registers in their indicated modes represent a 32-bit version of the 48-bit register in Figure 4-a. Both versions can perform the same operations with the only difference being register length. The 32-bit version can produce a rotate without affecting the right-most register, since it is by-passed electrically as is schematically shown. Figure 4-c shows three independent 16-bit registers, each isolated electrically from each other. In all three cases each register in the configuration can perform independent internal shifts without effecting the state of the other registers. For example, each can perform a rotate to the right. Collective as well as individual shifting is therefore possible without physically changing interconnections.

The configurations need not be strictly serial. Figure 5 shows two possible configurations. Figure 5-a is a series-parallel scheme. The parallel configuration in Figure 5-b can be thought of as a memory of four 16-bit words. If the four registers share common parallel data lines, data could be shifted about either parallel or serial.

Figure 6 shows the variable function register excluding control. While only three bits are shown, any number of bits could be used. With the aid of Figure

EXAMPLES OF THREE POSSIBLE CONFIGURATIONS
USING THREE IDENTICAL REGISTERS
TO FORM LARGER COMPOSITE REGISTERS



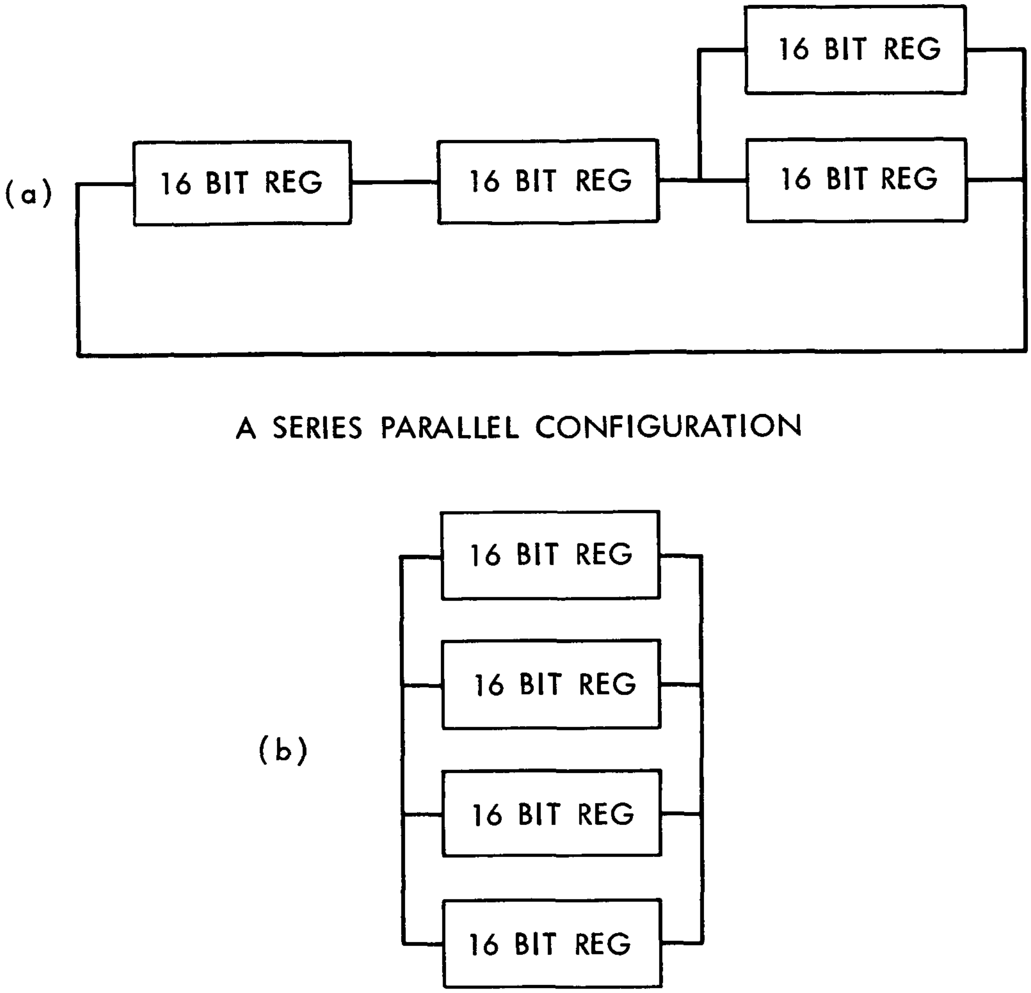
TOTAL OF $(8)^3$ DIFFERENT COMBINATIONS POSSIBLE

Figure 4

6, Table 1 and Table 2, the variable function register is completely described. Attention should be given to the square blocks which are labeled with letters. These are bidirectional switches which are used to transfer data into each register. Each operation of the register affects a data transfer, such as, a shift or load, by closing the appropriate gate. The square block labeled INV is an inverter.

The switches labeled COMP feed the inverted output of each stage back to its input. In this way the one's complement of the register is accomplished.

Clearing the register is accomplished by grounding the input leads to each stage through the switches labeled CLR.



A SERIES PARALLEL CONFIGURATION

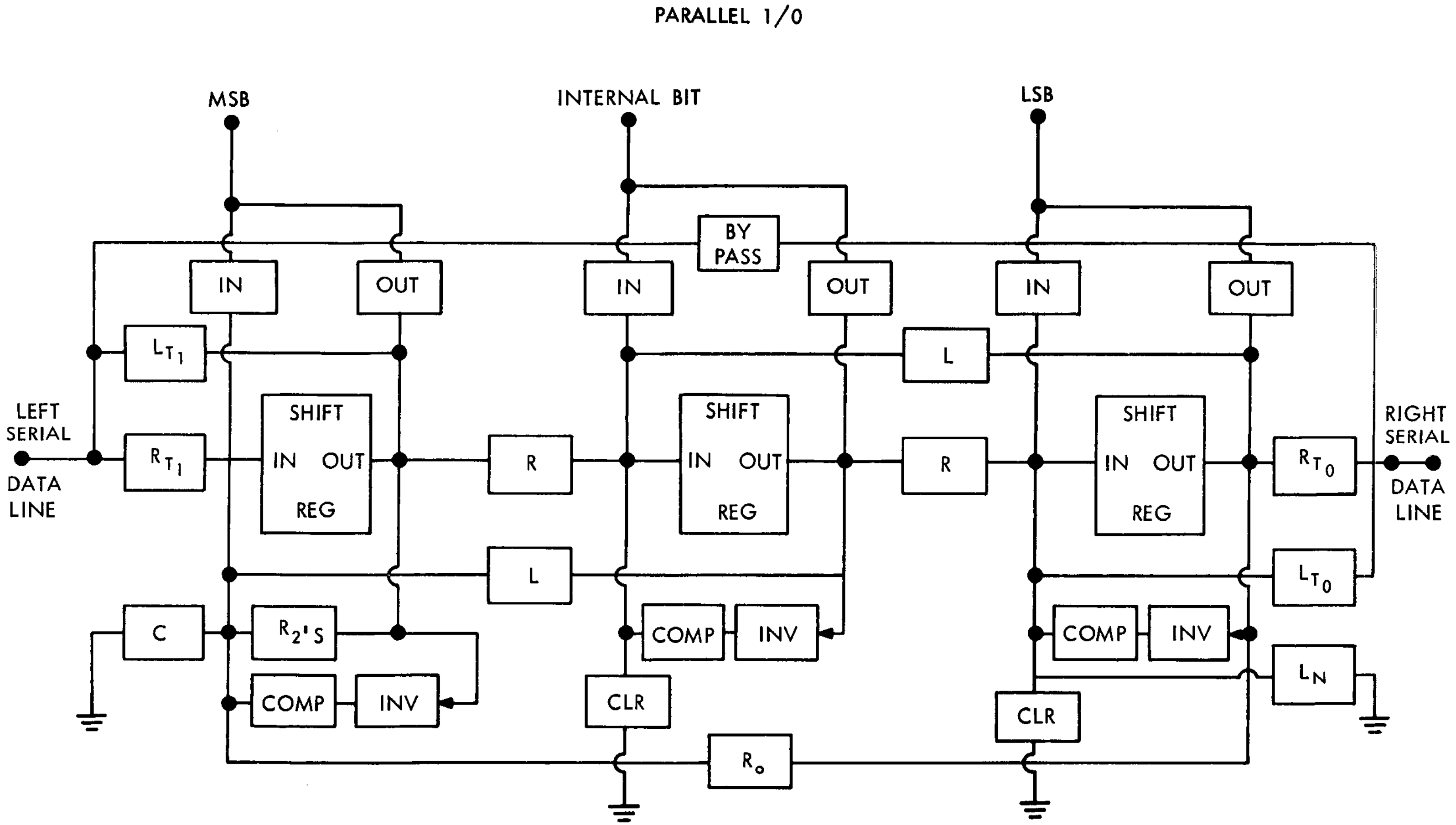
PARALLEL CONFIGURATION

Figure 5

In an effort to also minimize the lead count of the register system parallel data enters and leaves the register via the same lines. This is accomplished by insuring that the switches marked IN and OUT are never simultaneously on.

Because of the parallel nature of the one's complement, clear, input, and output operations, they function independent of mode.

Figure 7 shows the lead requirement of a 16-bit variable function register. As shown 18 leads are allocated for both serial and parallel transfer of data, four for the encoded instructions, three for the mode control, two for timing and two for power. The total lead requirement therefore is 29. The low lead



THREE STAGE MODEL OF THE
VARIABLE FUNCTION REGISTER

Figure 6

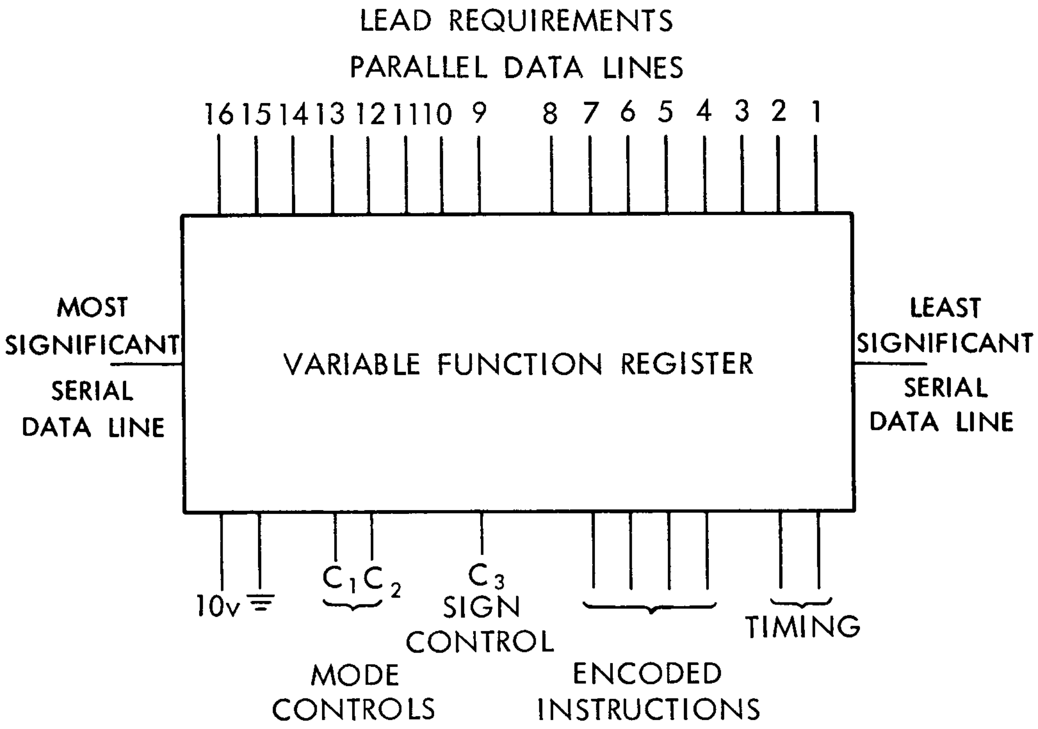


Figure 7

requirement is due to encoding all commands and instructions and using the serial/parallel data lines for bidirection transfer of data. As a point of clarification, 2 of the 3 leads for mode control are actually used to affect mode changes. The third lead is required to permit either zeroes or the contents of the most significant bit to propagate from left to right during either a total or non-total right shift.

FUTURE ARRAYS

Looking a bit into the future, we envision LSI arrays of greater capability as well as flexibility. Using the concept of modes and the Variable Function Register, designs are being completed on such an array. Known as the Parallel Processor, this device will not only perform the functions of the Variable Function Register but will also be able to perform in parallel such operations as two's complement addition, and subtraction, AND, OR and EXCLUSIVE OR. Just as many Variable Function Registers can be configured to form a composite system, the Parallel Processor will also have this ability.

CONCLUSION

To effectively use LSI technology, a closer look must be given to the design of small systems. It is both inefficient and costly to develop many large scale arrays that differ in only minor respects. As we continue to design and develop more sophisticated data processing systems the LSI arrays of tomorrow will be what the AND, OR and INVERTER circuits are today. While the latter three simple circuits have succeeded because of their basic inflexibility, the acceptable LSI arrays of the future must be multi-purpose, multi-function devices.

REFERENCES

1. M. M. Mitchell and R. W. Ahrons, 'MOS Micropower Complementary Transistor Logic," Computer Design, February, 1966.

2. R. W. Ahrons and P. D. Gardner, "Complementary MOS Integrated Circuits," Proceedings of Integrated Circuits Seminar, Stevens Institute of Technology, Hoboken, New Jersey, February 1, 1967.

3. H. R. Beelitz and H. S. Miller, "Partitioning for Large-Scale Integration," International Solid-State Circuits Conference, February, 1967.

4. J. R. Cricchi, "The Application of Complementary MOS Transistor Techniques to L.S.I.," University of Wisconsin, University Extension, Engineering Institutes, May, 1967.

Table 1
Switch Conditions for All Shift Instructions

Instruction	Mode	C_3	MSB Switches Conducting	Internal Bit Switches Conducting	LSB Switches Conducting	Other Switches Conducting Only In Mode 4
Non-Total Right	Any	0	$R_{2's}$	R	R	By-Pass
Non-Total Right	Any	1	C	R	R	By-Pass
Non-Total Left	Any	0	L	L	L_N	By-Pass
Non-Total Left	Any	1	L	L	L_N	By-Pass
Non-Total Rotate	Any	1	R_O	R	R	By-Pass
Total Right	1	0	R_{T1}	R	R	-
Total Right	1	1	R_{T1}	R	R	-
Total Right	2	0	$R_{2's}$	R	R, R_{TO}	-
Total Right	2	1	C	R	R, R_{TO}	-
Total Right	3	0	R_{T1}	R	R, R_{TO}	-
Total Right	3	1	R_{T1}	R	R, R_{TO}	-
Total Right	4	0	$R_{2's}$	R	R	By-Pass
Total Right	4	1	C	R	R	By-Pass
Total Left	1	0	L	L	L_{TO}	-
Total Left	1	1	L	L	L_{TO}	-
Total Left	2	0	L_{T1}	L	L_N	-
Total Left	2	1	L_{T1}	L	L_N	-
Total Left	3	0	L_{T1}	L	L_{TO}	-
Total Left	3	1	L_{T1}	L	L_{TO}	-
Total Left	4	0	L	L	L_N	By-Pass
Total Left	4	1	L	L	L_N	By-Pass
Total Rotate	1	Any	R_{T1}	R	R_{TO}	-
Total Rotate	2	Any	R_{T1}	R	R_{TO}	-
Total Rotate	3	Any	R_{T1}	R	R_{TO}	-
Total Rotate	4	Any	-	-	-	By-Pass

Table 2
Logic Equation for All Switches

Note: Symbols on left side of equal sign represent switches in Figure 6. Symbols other than "P" on the right side of equal sign represent instructions and mode commands.

The switches in Figure 6 are permitted to conduct at a time indicated by "P".

Let P = Timing pulse which activates the appropriate transfer switches

$R = (R_0 + R) \cdot P$

$L = L \cdot P$

$R_{TO} = [R \cdot T \cdot C_2 + R_0 \cdot T \cdot (C_1 + C_2)] \cdot P$

$L_{TO} = L \cdot T \cdot C_2 \cdot P$

$L_N = (L \cdot \overline{T} + L \cdot T \cdot \overline{C}_2) \cdot P$

$R_{T1} = [R \cdot T \cdot C_1 + R_0 \cdot T \cdot (C_1 + C_2)] \cdot P$

$L_{T1} = L \cdot T \cdot C_1 \cdot P$

$R_{2's} = (\overline{T} \cdot \overline{C}_3 \cdot R + R \cdot T \cdot \overline{C}_1 \cdot \overline{C}_3) \cdot P$

$C = (\overline{T} \cdot C_3 \cdot R + R \cdot T \cdot \overline{C}_1 \cdot C_3 + CLR) \cdot P$

$R_0 = R_0 \cdot \overline{T} \cdot P$

$IN = IN \cdot P$

$OUT = OUT \cdot P$

$COMP = COMP \cdot P$

$BY\text{-}PASS = \overline{C}_1 \cdot \overline{C}_2$ (Independent of time)

$CLR = CLR \cdot P$